2006

Streak Detection in Mottled and Noisy Images

Hector Santos

STREAK DETECTION IN MOTTLED AND NOISY IMAGES

By

Hector Santos

A Thesis submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTERS OF SCIENCE

In

Electrical Engineering

Approved by:

Professor Eli Saber
(Dr. Eli Saber – Advisor)

Professor Sohail Dianat
(Dr. Sohail A. Dianat – Committee Member)

Advisor Wencheng Wu
(Dr. Wencheng Wu – Committee Member)

Professor Vincent Amuso
(Dr. Dr. Vincent Amuso – Department Head)

DEPARTMENT OF ELECTRICAL ENGINEERING

COLLEGE OF ENGINEERING

ROCHSTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

SEPTEMBER 2006

THESIS RELEASE PERMISSION

DEPARTMENT OF ELECTRICAL ENGINEERING

COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

Title of Thesis:

STREAK DETECTION IN MOTTLED AND NOISY IMAGES

I, Hector Santos, hereby grant permission to Wallace Memorial Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature Hector Santos

ACKNOWLEDGMENTS

First I must thank Dr. Eli Saber. Not only because he taught me everything that I know in digital signal processing but more importantly because he taught me how to research. I would like to thank Dr. Wecheng Wu, and Dr. Sohail A. Dianat for taking time out of their busy schedule and going through my thesis, providing their valuable insights and constructive criticism.

This work was supported by the Center for Electronic Imaging Systems, a NYSTAR-designated Center for Advanced Technology, Xerox Corporation and the Electrical Engineering Department at RIT.

ABSTRACT

This thesis describes an algorithm for detecting streaks in printed images using adaptive window-based image projections and maximization of mutual information. To this effect, projections are computed across the entire image at different window sizes. The traces collected from the projections are correlated using maximization of mutual information to pinpoint streak locations and widths using a peak detection algorithm. Finally, for a given peak, the window size is changed adaptively to identify and locate the intensity and length of the corresponding streak while maximizing signal to noise ratio. Results on synthetic and real-life images are provided to demonstrate the effectiveness of the proposed technique.

Keywords: streak, pixel projection, adaptive window size, maximization of mutual information.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS....3
ABSTRACT....4
Table of Contents....5
List of Figures....6
List of Tables....7
Chapter 1....8
1 Introduction....8
Chapter 2....11
2 Data Analysis....11
Chapter 3....13
3 Proposed Algorithm....13
3.1 Halftone screen filtering....15
3.2 Computation of horizontal and vertical projections....16
3.3 Correlation of projections using MMI....20
3.3.1 Correlation of traces....23
3.3.2 Computation of the confidence vector....24
3.4 Computation of length and intensity of streaks....25
Chapter 4....29
4 Results....29
4.1 Analysis on synthetic images....30
4.2 Analysis on real images....33
Chapter 5....39
5 Conclusion....39
References....41

TABLE OF FIGURES

Figure 1: An example of mottle 12

Figure 2: An example of Halftone 12

Figure 3: An example of Streak 12

Figure 4: Block Diagram of the proposed Streak Detection Algorithm 14

Figure 7: Projection Computation 17

Figure 8: (a) Image size = 128 x 128, (b) Vertical Projection, window size = 128 x 128, (c) Vertical Projection, window size = 32 x 128 18

Figure 9: (a) Vertical Projection with Intensity Gradient, (b) Average Vertical Projection, (c) Vertical Projection without Intensity Gradient 19

Figure 10: Average Filter, (a) Impulse response, (b) Frequency response 20

Figure 11: Spatial Correlation of Vertical Projections 22

Figure 12: The Confidence Vector 23

Figure 13: Technique 1 Confidence V ector, Technique 2 Confidence Vector 25

Figure 14: (a) Non Periodic Vertical Confidence Vector, (b) Periodic Vertical Confidence Vector Fourier transform, (c) Periodic Vertical Confidence Vector. 26

Figure 15: Determination of Length of the streak, (a) Spanning window, (b) Initial Vertical Position Profile, (c) Final Position Profile. 28

Figure 16: Synthetic image with streaks of different length and intensity, (b) Detected streaks for $\sigma^2 = 20$, (c) Confidence Vector. 30

Figure 17: (a) Synthetic image with streaks of different width and intensity, (b) Detected streaks for $\sigma^2 = 20$, (c) Confidence Vector. 31

Figure 18: (a) Synthetic image with streaks of different intensity, (b) Detected streaks for $\sigma^2 = 20$, (c) Confidence Vector 32

Figure 19: ROC analysis curve 33

Figure 20: Results on Real Images 37

LIST OF TABLES

Table 1: Length vs Noise Variance 31

Table 2: Width vs Noise Variance 32

Table 3: Intensity vs Noise Variance 33

Table 4: Optimum Thresholds 34

Table 5: Length 38

Table 6: Width 38

Table 7: Intensity 38

Chapter 1

1. INTRODUCTION

The standards for image quality have increased significantly over the past decade, and will continue to do so for years to come. Present day print engines are required to meet consistent and stable image quality requirements as measured by various metrics and ultimately evaluated by customers. Generally speaking, print shops place image quality as one of the most important aspects of any printing system. However, even though the quality of the documents produced using current print engines far exceeds what was generated a decade or two ago, the current devices still possess a variety of image quality defects and artifacts (e.g. spots, streaks, etc.) that often result from a fault or degradation in the underlying imaging and electrophotographic processes (EP). The artifacts come in a variety of sizes and shapes and occur at different spatial locations. Operator or engineer's intervention is usually required to visually or instrumentally diagnose the

defect and perform corrective action as quickly as possible to minimize downtime. This requires expertise and knowledge in EP and image quality. Hence, the need for an automated algorithm for detecting and diagnosing defects with minimal intervention.

Defect analysis algorithms have been explored in medical and EP literature. To this effect, pixel projection is a well known method used to convert a two dimensional problem into a one dimensional problem. This method is the basis of computed tomography scans. Computed tomography (CT) can be defined as the reconstruction of sections of an object from image projections taken at different angles [1]. The applications of CT are diverse: medical imaging, airport security, and non-destructive testing in manufacturing. Noise-induced streak artifacts arise in X-ray CT when some or all of the transmission measurements capture relatively few photons due to the attenuation along the projection measurement lines [2]. The high variability of these photon-starved measurements is amplified by the logarithm and amp-filtration steps employed for reconstruction; and the resulting inconsistencies produce streaks in the reconstructed image [2]. The conventional approach to noise control in CT is a shift invariant filtration of all projections. Smoothing the photon starved measurements to eliminate streak artifacts usually results in a degradation of image resolution. More successful strategies have involved the use of adaptive filters that attempt to adjust the degree of smoothing to the local noise levels in the measurements [2, 3]. Riviere and Billmire [4] presented a statistical sinogram smoothing approach for x-ray CT with the intention of reducing noise-induced streak artifacts.

On the other hand, there have been several studies related to analyzing streaks in EP. To this effect, organic photoconductive drum velocity perturbations that cause line spacing

variations have been shown [5-7] to be significant contributors to the generation of streak artifacts across a printed page perpendicular to the process direction in EP. Chen and Chiu [8] proposed an improved regulation of the organic photoconductive drum rotational velocity in order to improve EP stability and reduce banding artifacts. Rasmussen [9] discusses the quantification of image quality in printed documents that contain streaks, bands, and mottle and shows how several measurements techniques can be combined to yield effective metrics. However, none of the above techniques analyze the aforementioned streaks to yield their length, position, width, and intensity.

In this thesis, I propose an automated algorithm for detecting the location of vertical and horizontal streaks in printed documents and assessing their corresponding length, width, and intensity. The propose algorithm uses adaptive window-based image projections and maximization of mutual information (MMI) [10] for detecting "straight line" streaks in noisy and mottled grayscale and RGB color images. The traces, collected from various window sizes, are correlated using MMI to build a confidence vector that indicates the location and width of the streak(s). Finally, for a given peak position, the window size is changed adaptively to determine the intensity and length of the corresponding streak thereby maximizing the underlying signal to noise ratio. The effectiveness of the proposed algorithm is demonstrated on a series of images with varying degrees of streak intensity, length and width. The strength of the proposed algorithm lies in its ability to effectively automatically detect and localize streaks in noisy images and compute their corresponding length, width and intensity attributes.

Chapter 2

2.0 DATA ANALYSIS

In oder to familiarize the reader with the data set on which the algorithm was tested, the rest of this section presents some definitions and specifications that will be useful for a better understanding of this thesis.

The data set is formed of images in RGB format. These samples were scanned at 600 dpi and presented the following image quality defects: mottle, halftone, and streak.

Mottle

It is a term used to describe the non-uniformity of a printed document. It gives an image a textured and grainy appearance.

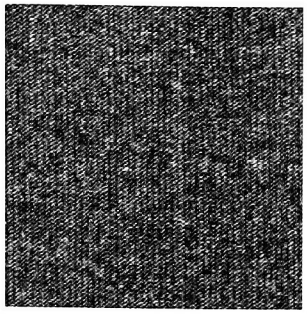


Figure 1: An example of mottle

Halftone

Halftones are produced by photographing an image through a screen. The screen frequency determines how many dots are used to make each spot or gray.

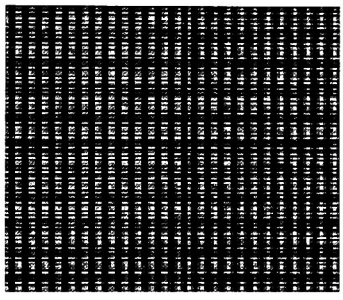


Figure 2: An example of Halftone

Streak

Streaks are straight lines that appear in printed documents. Organic photoconductive drum velocity perturbations that cause line spacing variations have been shown to be significant contributors to the generation of streak artifacts across a printed page perpendicular to the process direction in electrophotographic processes

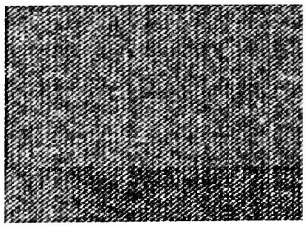


Figure 3: An example of Streak

Chapter 3

3. PROPOSED ALGORITHM

A block diagram of the proposed algorithm is shown in Figure 4. It is divided into four major steps. The first step is dedicated to identifying and de-screening the given image in order to minimize the impact of the halftone screen on the proposed algorithm. In Step 2, horizontal and vertical projections are computed using varying window sizes for the image at hand. In Step 3, one-dimensional profiles are correlated using MMI yielding a confidence vector that serves to indicate the location and width of the streak(s). Finally, in Step 4, the length and intensity of each streak are calculated using an adaptive window size selection technique to maximize the signal to noise ratio (SNR). An example image is also shown for illustration purposes. The steps of the algorithm are discussed in detail in the following paragraphs.

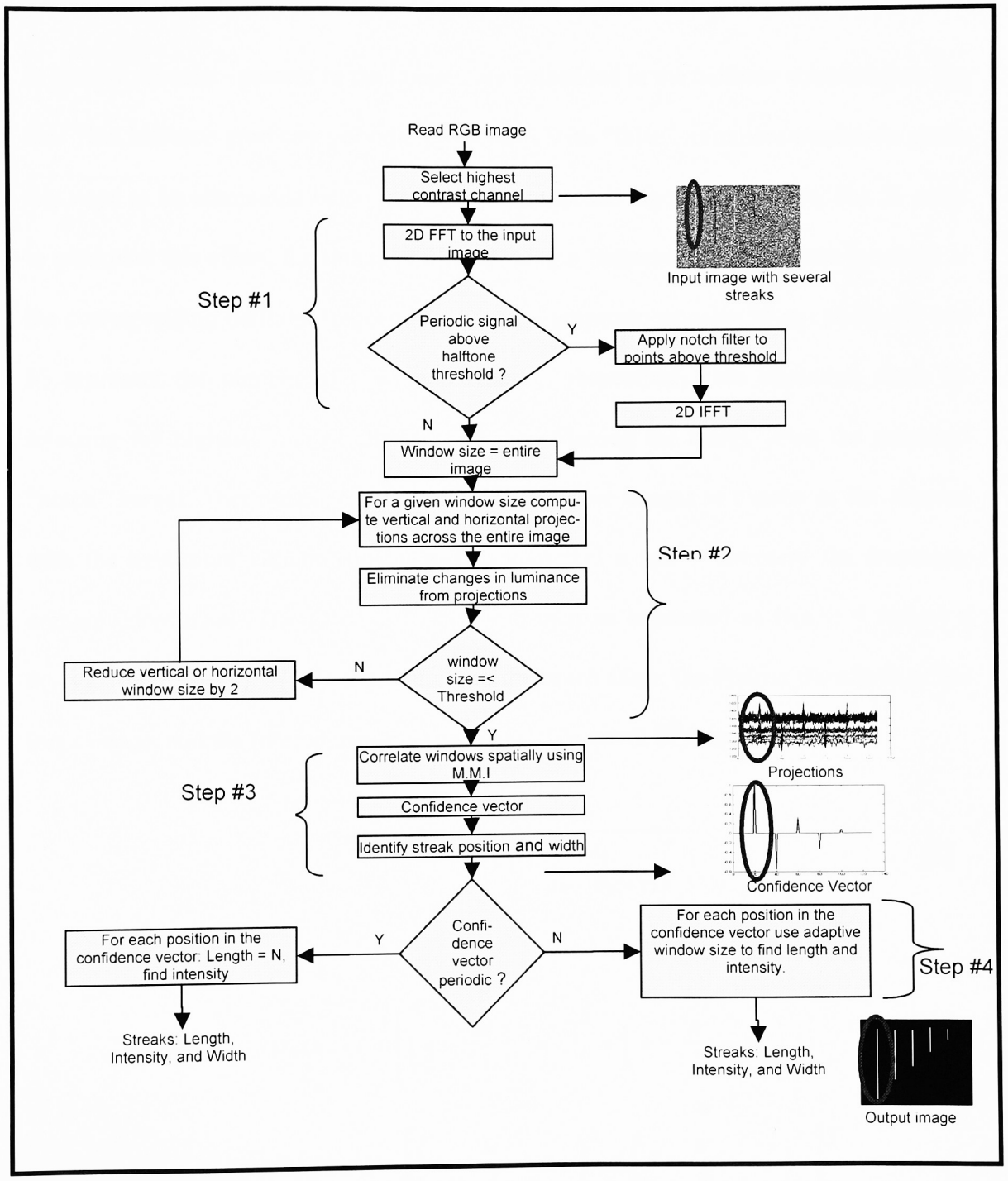


Figure 4: Block Diagram of the proposed Streak Detection Algorithm

3.1 Halftone screen filtering

In general, streaks - present in the image - are embedded in the halftone structure (see Fig 5a). This halftone produces periodic projections with "false" minimum/maximum peaks that need to be eliminated before applying the proposed algorithm (see Fig 5b). In order to minimize this effect, the image is filtered using a Butterworth notch filter localized at the corresponding halftone frequencies. Given a grayscale or color image where N_1 and N_2 represent the number of rows and columns respectively, the algorithm starts by selecting the highest contrast channel and then filtering the image using the proposed "notch" kernel. The highest contrast channel for color images is chosen as the channel with the maximum variation in intensity values. Let u and v represent the frequency domain coordinates. The filtering process can then be expressed as G(u,v) = H(u,v) x F(u,v), where H(u,v), F(u,v) and G(u,v) are the notch filter, the Fourier transform of the input image, and the filtered image respectively. The notch filter is defined as:

$$H(u,v) = \frac{1}{1+\left[\frac{D^2_0}{D_1(u,v)D_2(u,v)}\right]^n} \tag{1}$$

$$D_1(u,v) = \left[\left(u - \frac{M}{2} - u_0\right)^2 + \left(v - \frac{N}{2} - v_0\right)^2\right]^{\frac{1}{2}} \tag{2}$$

$$D_2(u,v) = \left[\left(u - \frac{M}{2} + u_0\right)^2 + \left(v - \frac{N}{2} + v_0\right)^2\right]^{\frac{1}{2}} \tag{3}$$

D_1 and D_2 represent the "notch" locations, D_o the radius of the filter, and n its corresponding order. An example is shown in Figure 6. It should be noted that the proposed algorithm considers any structure above 150 dpi to be part of the halftone screen and as such is filtered from the streak detection process.

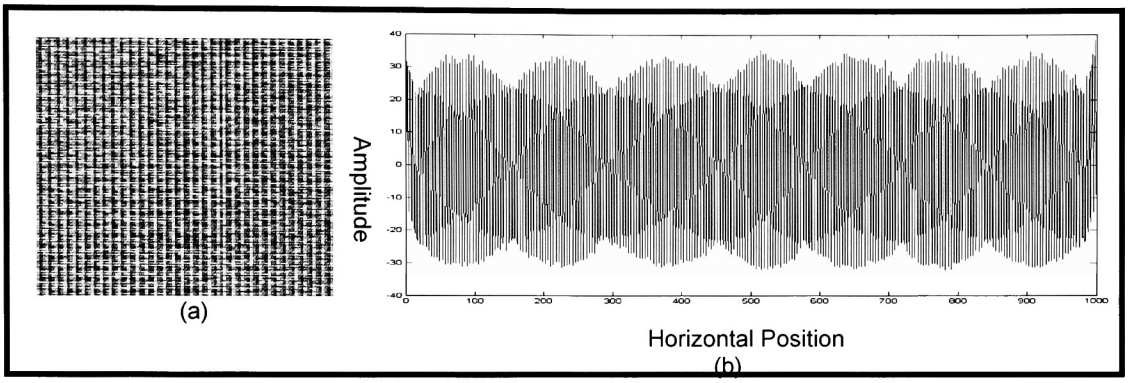


Figure 5: (a) Halftone, (b) Periodic Vertical Projections

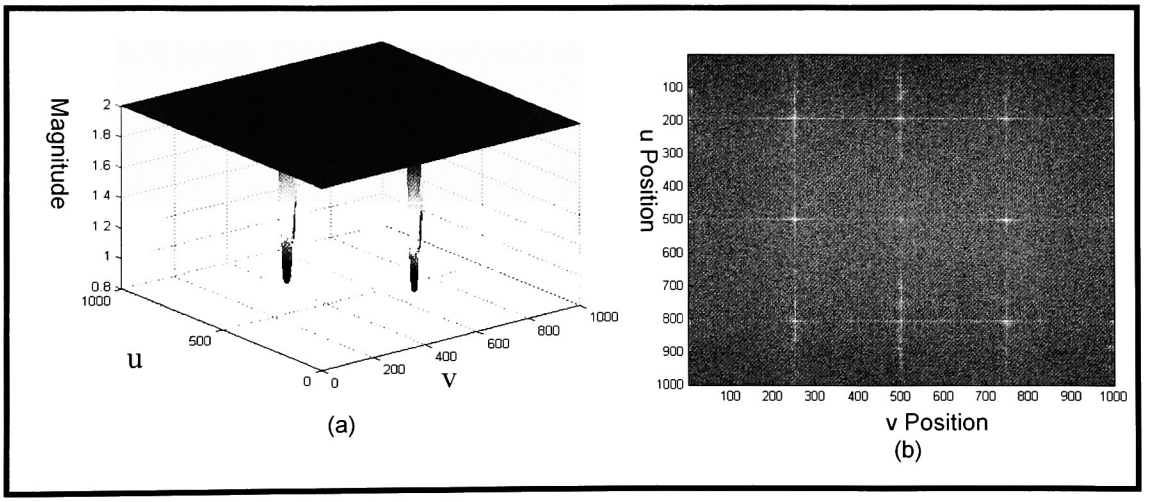


Figure 6: (a) Magnitude Response for the Notch filter, (b) F. T of the image shown in Fig. 5a

3.2 Computation of Horizontal and Vertical Projections

Once the image has been filtered to minimize the effect of the halftone screen on the streak detection process, the algorithm computes vertical projections for a window of size $M_1 \times M_2$ initially chosen to equal the entire image (M_1 and M_2 represent the number of rows and columns respectively). To this effect, let $X(i,j)$ represent the intensity value at location (i, j). A vertical projection is then defined as the average intensity value of all pixels for a given column bounded by the window size. The window size is then reduced vertically by a factor of 2, and projections are computed using the $M_1/2 \times M_2$ window in a sliding fashion starting from the top of the image and moving downward in steps equal to $M_1/4$ (i.e. half of the window size in the vertical direction). This process is again repeated using an $M_1/4 \times M_2$ window size and continuing until the number of rows in the window is less than 8 pixels. This can be expressed mathematically as:

$$P_{(K_1,K_2;M_1,M_2)}(j) = \frac{1}{M_1} \sum_{i=K_1}^{K_1+M_1-1} X(i,j) \qquad (4)$$

where K_1 and K_2 represent the start location of the window as shown in Figure 7

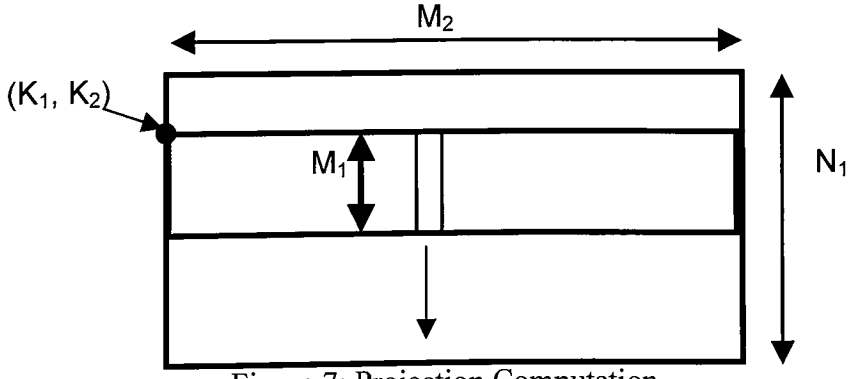


Figure 7: Projection Computation

$$P_{M_1,M_2}^{K_1,K_2}(j)$$

The reason for using different window sizes to detect different streaks length is based on the fact that the signal to noise ratio is increased significantly as the window size gets closer to the length of the streak. Fig. 8a shows a synthetic image (128 × 128) with several streaks of different lengths and intensity where the average background value has been set to 128. Gaussian noise with variance equal to 10 was added to the synthetic image. The red and blue rectangles indicate the selected 128 × 128 and 32 × 128 windows used to compute the vertical projections shown in Fig 8b and 8c respectively. Note that the signal to noise ratio for the smallest streak (see Fig. 8) is significantly increased as the window size is reduced from 128 to 32 in the vertical direction.

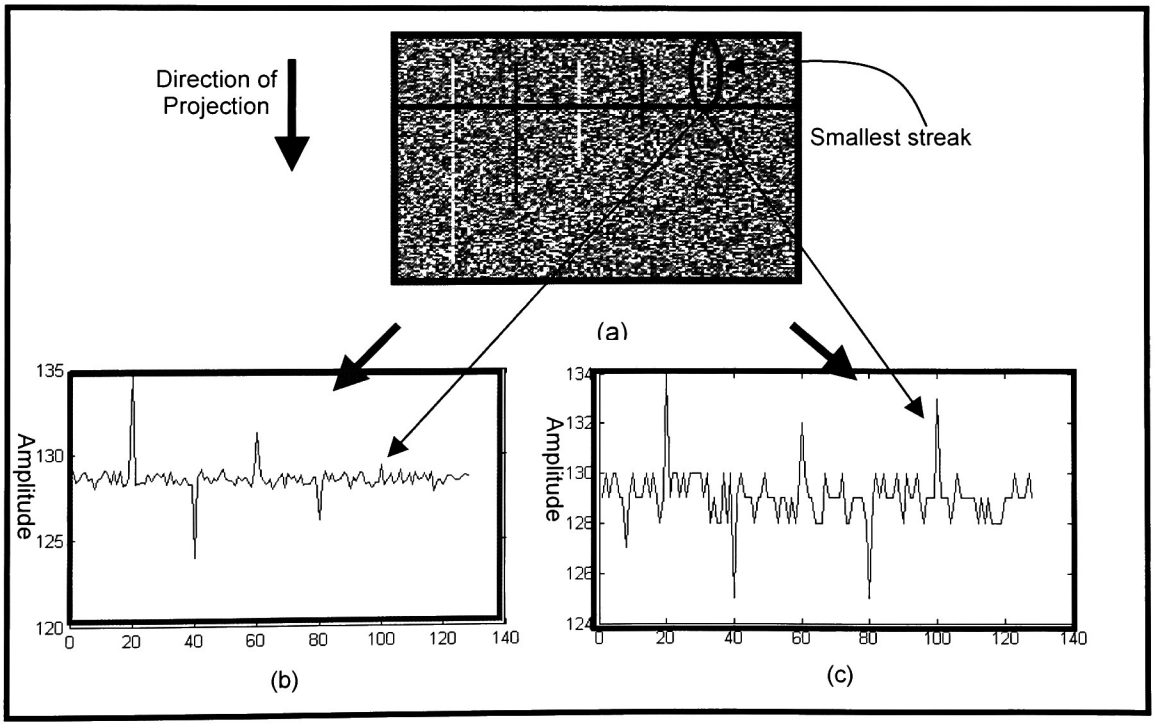


Figure 8: (a) Image size = 128 × 128, (b) Vertical Projection, window size = 128 × 128, (c) Vertical Projection, window size = 32 × 128

Low frequency changes in intensity across the image produce "false" increases/decreases in projection values yielding false peaks that need to be minimized prior to streak detection. Let x[n] denote a typical vertical projection profile as shown in Figure 9a for a given image. Hence, our streak algorithm utilizes a moving average filter h[n] (see Figure 10) to reduce irregularities and random variations in the signal. The order of the filter was determined empirically as L=30. The output of the filter $x_f[n]$ is shown in Figure 9b, and the corrected projection profile y[n] is displayed in Figure 9c.

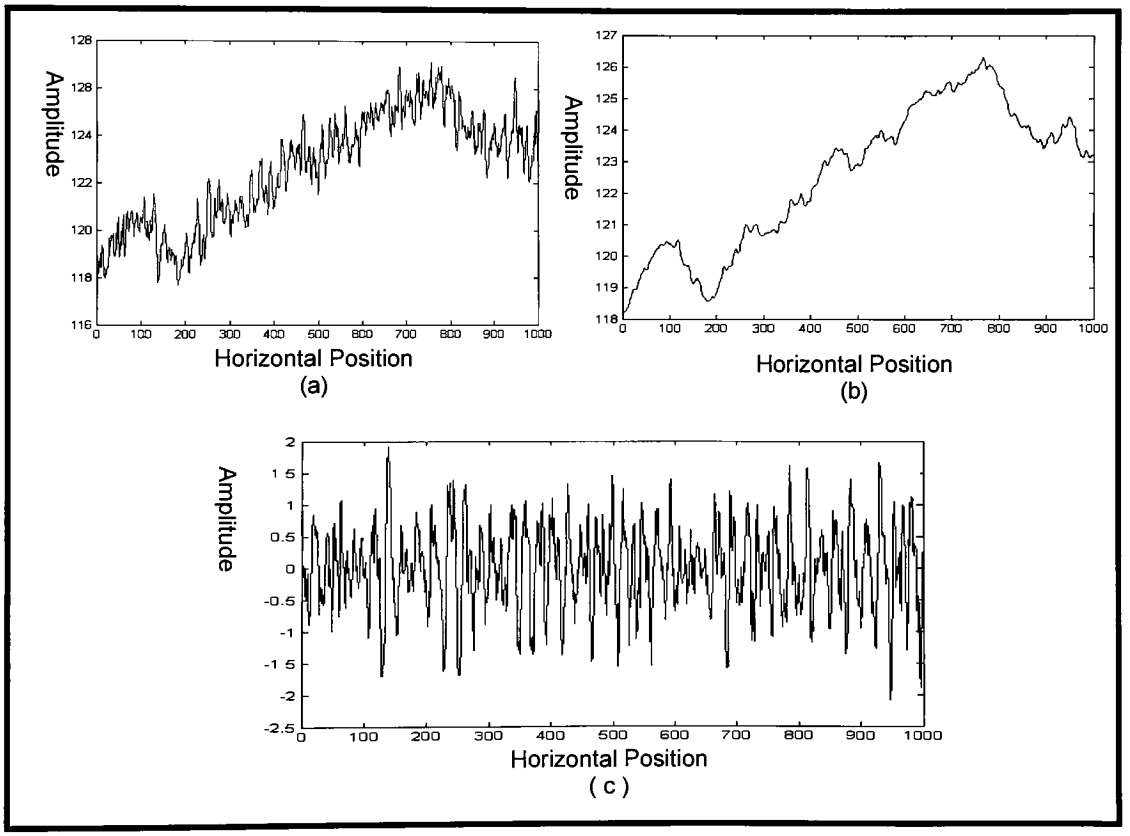


Figure 9: (a) Vertical Projection with Intensity Gradient, (b) Average Vertical Projection, (c) Vertical Projection without Intensity Gradient.

The corrected projection was obtained by filtering the input signal shown in Figure 9a and subtracting the filtered version (see Fig. 9b) from the original (see Fig. 9a) yielding

the signal shown in Fig. 9c. In other words, y[n] = x[n] – x[n] * h[n] where * indicates convolution [11]. The above intensity corrected projections are utilized collectively to generate the confidence vector as described in the following section in order to pinpoint the location and width of the streak(s). Similarly, the above process is also repeated in the horizontal direction yielding horizontal type projections.

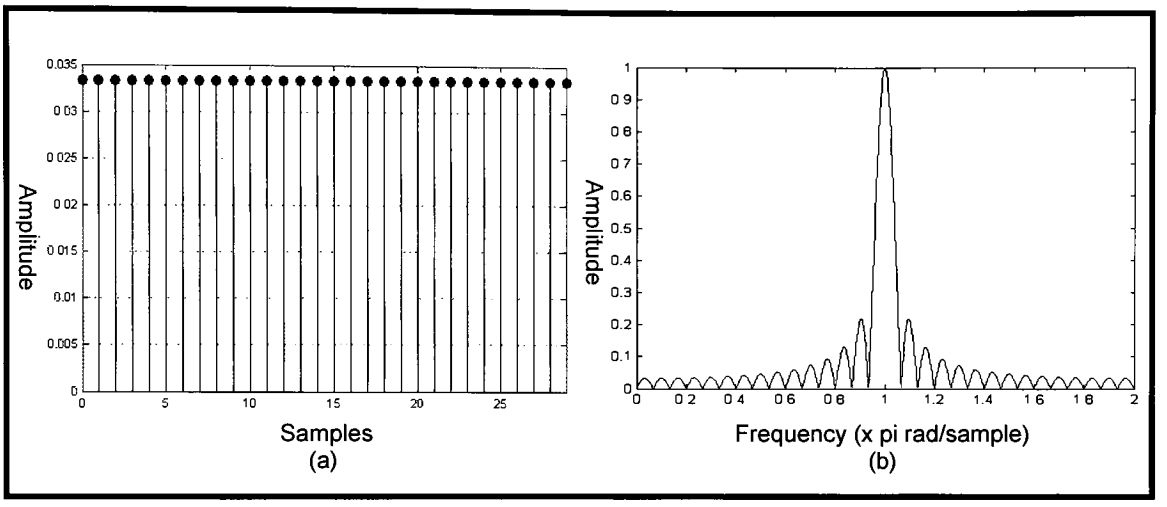


Figure 10: Average Filter, (a) Impulse response, (b) Frequency response

3.3 Correlation of Projections using MMI

Once the projections have been collected from all window sizes as described in Section 3.2, we proceed to generate the confidence vector using MMI [10, 12] in order to pinpoint horizontal and vertical streak locations. Let x and y define two random variables. The mutual information, which is a measure of general interdependence between random variables [13], is defined as:

$$I(x,y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{5}$$

where *p(x)* and *p(y)* are the marginal probability density functions (pdf), and $p(x,y)$ is the joint pdf of the random variables *x* and y. The corresponding entropies are defined as:

$$H(x_i) = -\sum_{i=1}^{Q_1} P(x_i)\log\, p(x_i) \tag{6}$$

$$H(y_i) = -\sum_{j=1}^{Q_2} P(y_i)\log\, p(y_i) \tag{7}$$

$$H(x_i, y_i) = -\sum_{j=1}^{Q_2}\sum_{i=1}^{Q_2} P(x_i, y_i)\log\, p(x_i, y_i) \tag{8}$$

where Q1 and Q2 represent the number of states for the variables x and y. Hence, the mutual information between the two random variables *x* and *y* can be expressed in terms of the entropies [13] as:

$$I(x,y) = H(x) + H(y) - H(x,y) \;\geq 0 \tag{9}$$

It is zero if x and y are statistically independent and increases monotonically as x and y become more statistically dependent.

The projections collected above from the various window sizes are correlated using MMI [10] to yield a horizontal and vertical confidence vector that indicates the location and width of the streak(s) in the image for both horizontal and vertical directions, respectively. To this effect, for a given direction (i.e. horizontal or vertical), we correlate the traces by computing the mutual information between detected peaks at different window sizes. Fig. 11 illustrates the correlation among various vertical projections taken at different windows sizes ranging from 128 × 128 to 8 × 128 for a synthetic noise free 128 × 128 image that contains several streaks. The windows and their corresponding projections are clearly indicated in the Figure using different colors. Note the horizontal correspondence in the peak locations for the different windows. On the other hand, the

height of the peak in the projection profile for a given streak changes drastically as a function of the window size as shown for the smallest streak in Fig. 11.

The number of projections needed by the algorithm to compute the position, width, length, and intensity of the streaks is defined by:

$$W_n = \sum_{i=1}^{l} (2^i - 1) \qquad (10)$$

where w_n represents the number of windows needed by the algorithm and l the number of decomposition levels. In Figure 11, the decomposition levels are represented by 128, 64, 32, 16, and 8.

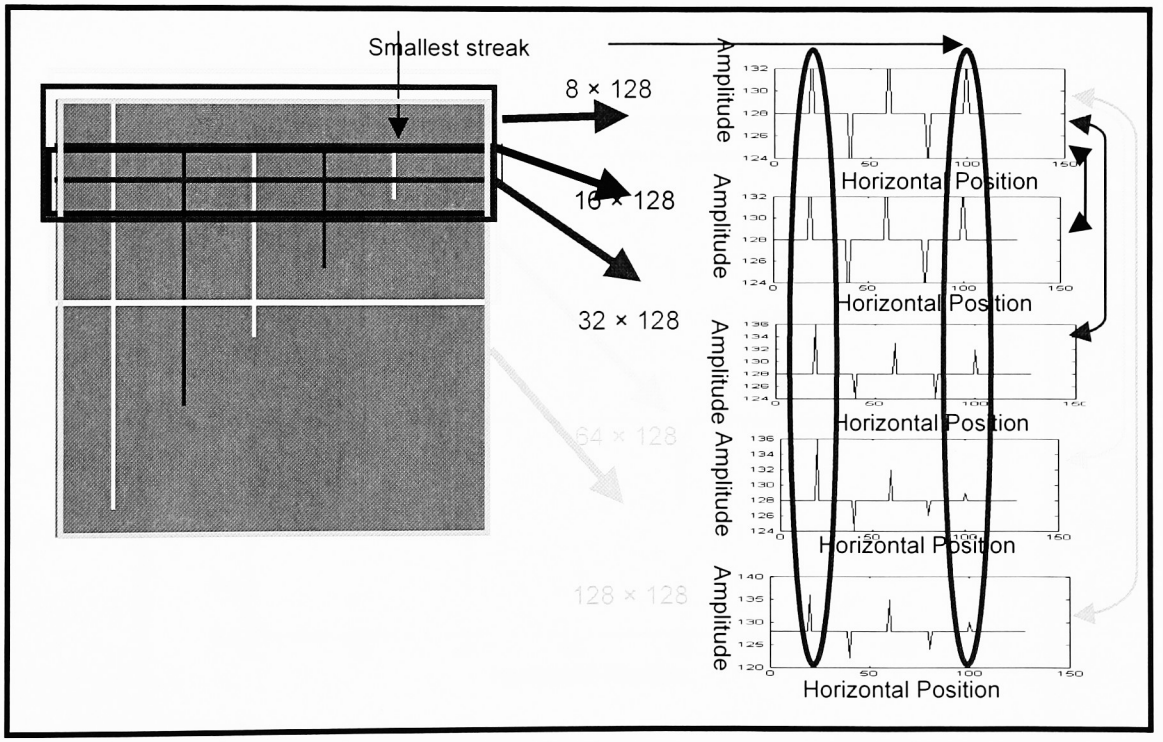


Figure 11: Spatial Correlation of Vertical Projections
Some windows are drawn bigger for better visualization

The spatial correlation is carried out as follows:

1. Detect peaks on the smallest window size $M_{\min} \times N$, where $M_{\min}$ is the minimum number of rows and N is the number of columns of the image.
2. Compute $I_i(X_i,Y_i)=H(X_i)+H(Y_i)-H(X_i,Y_i)$, where X_i is a peak in the smallest window size at position i and Y_i is a peak at position i at a larger window size.
3. If $I_i(X_i,Y_i)$ is greater than a user specified threshold T then assign k to that location. k is defined as one divided by the total number of projections.
4. Repeat 2 and 3 until Y_i belongs to the largest window size.
5. Repeat 1 through 4 for each projection from the smallest window size.

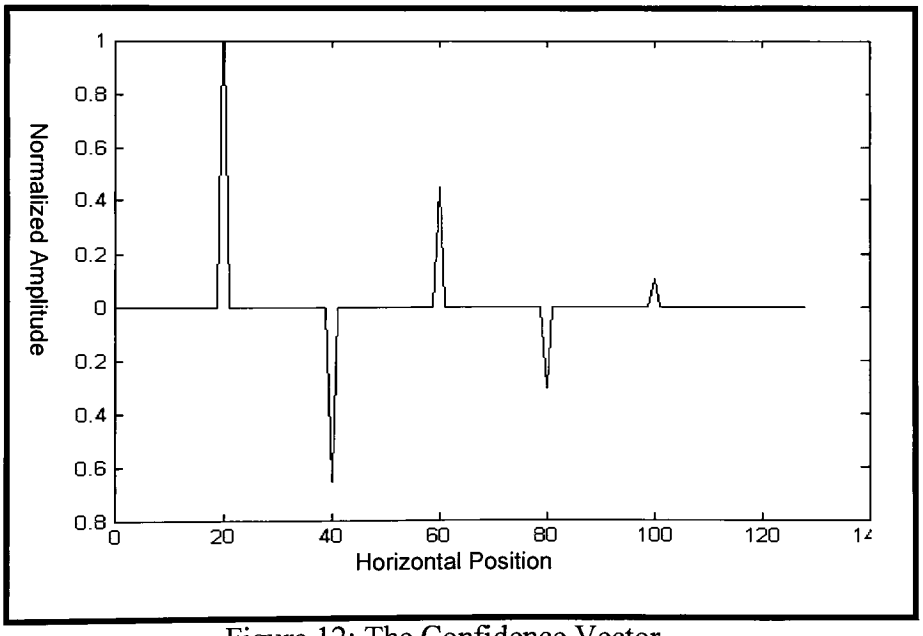


Figure 12: The Confidence Vector

Locations that exhibit a confidence value greater than a user specified threshold are selected as streak(s). The width of the streak is defined by the width of the peak in the confidence vector (see Fig. 12).

3.3.1 Correlation of Traces

I propose two techniques to correlate the traces collected from different window sizes to obtain a confidence vector: 1) In Technique 1, for a given window size and direction (i.e. horizontal or vertical), correlate traces by computing the mutual information between peaks in two consecutive projections to obtain an intermediate confidence vector. Then correlate the intermediate confidence vectors from the various window sizes to obtain a final confidence vector. Locations that exhibit a confidence value greater than a user specified threshold are selected as streak(s).

2) In Technique 2, I used correlation of traces from windows that spatially contain streak (s), see Fig 11. For a given location and direction (horizontal or vertical) I correlate the traces by computing the mutual information between peaks from the smallest window size projection and each of the projection from larger window sizes. Locations that exhibit a confidence value greater than a user specified threshold are selected as streak(s). In these experiments, the spatial correlations of traces as discussed in Technique 2, offered better results than Technique 1. This is due to the fact that: 1) Technique 1: correlates all the traces for a given window size. If the streak does not extent across the entire image there is no need to find the correlation between all of them when some of the traces do not contain information about the streak. This makes the algorithm slower. 2) Technique 2: offers better detection for small streaks because it is less sensitive to random alignment of noise that might be seen as "streaks" by the algorithm. I applied the same input image Fig. 13a (with a noise variance of 35) to both techniques (spatial correlation of traces and correlation of all traces for a given window size). Fig 13b and

13c show the output confidence vector from each technique. As we can see in Fig 13b small streak detection is not possible due to peaks caused by random alignment of noise.

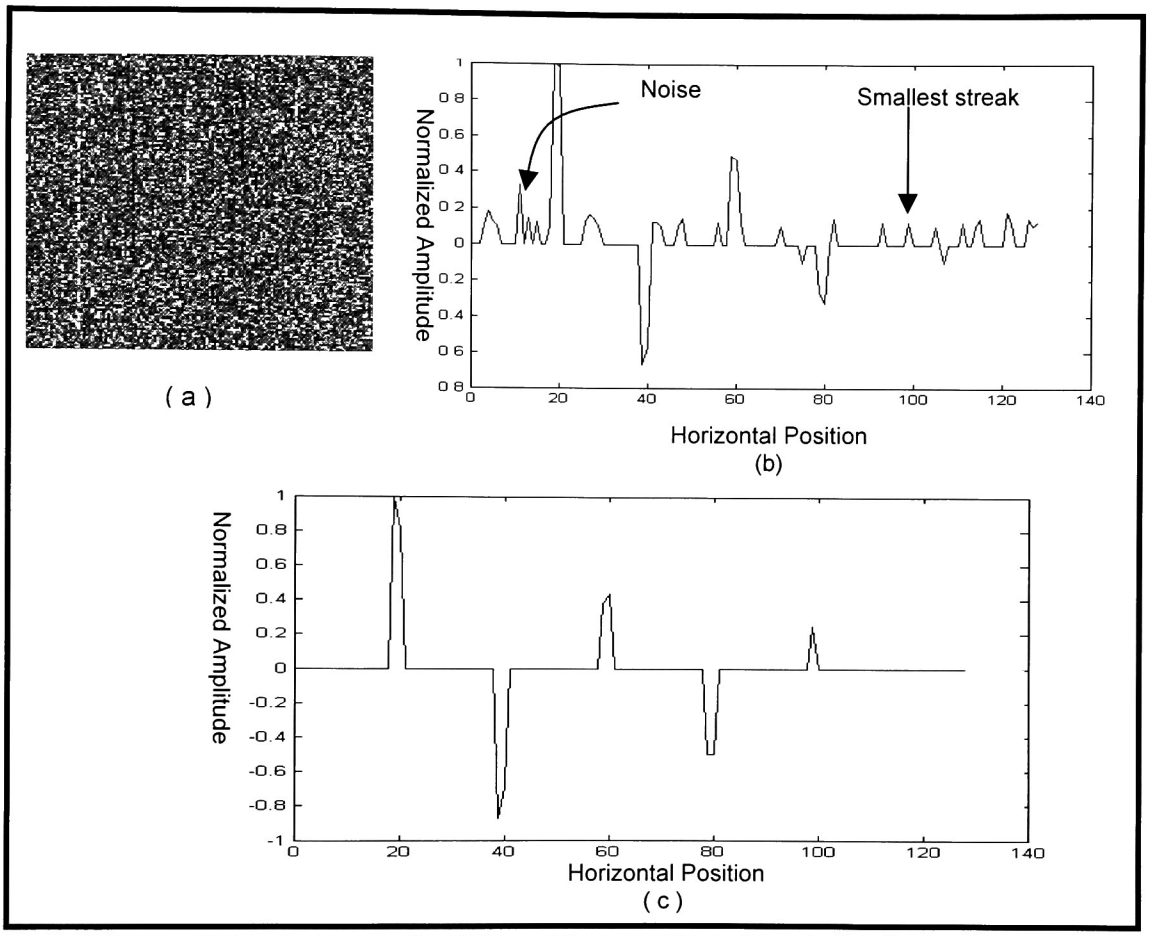


Figure 13: Technique 1 Confidence V ector, Technique 2 Confidence Vector

3.3.2 Computation of the Confidence Vector

The confidence vector shows the mutual information contained in the projections obtained from different window sizes. The peaks in the confidence vector provide information about the horizontal position and width of the streaks for a specified threshold. The longer the streak, the higher the peak in the confidence vector due to more projections contains information about the streak.

Once I have the confidence vector, there are have two possible streak patterns: periodic and non-periodic. I apply the Fourier transform on the confidence vector to determine its periodicity. This algorithm utilizes a peak detection algorithm in the frequency domain to find the period(s) of the signal(s). Fig. 14 shows a periodic and non-periodic confidence vector and the Fourier transform for the periodic pattern.

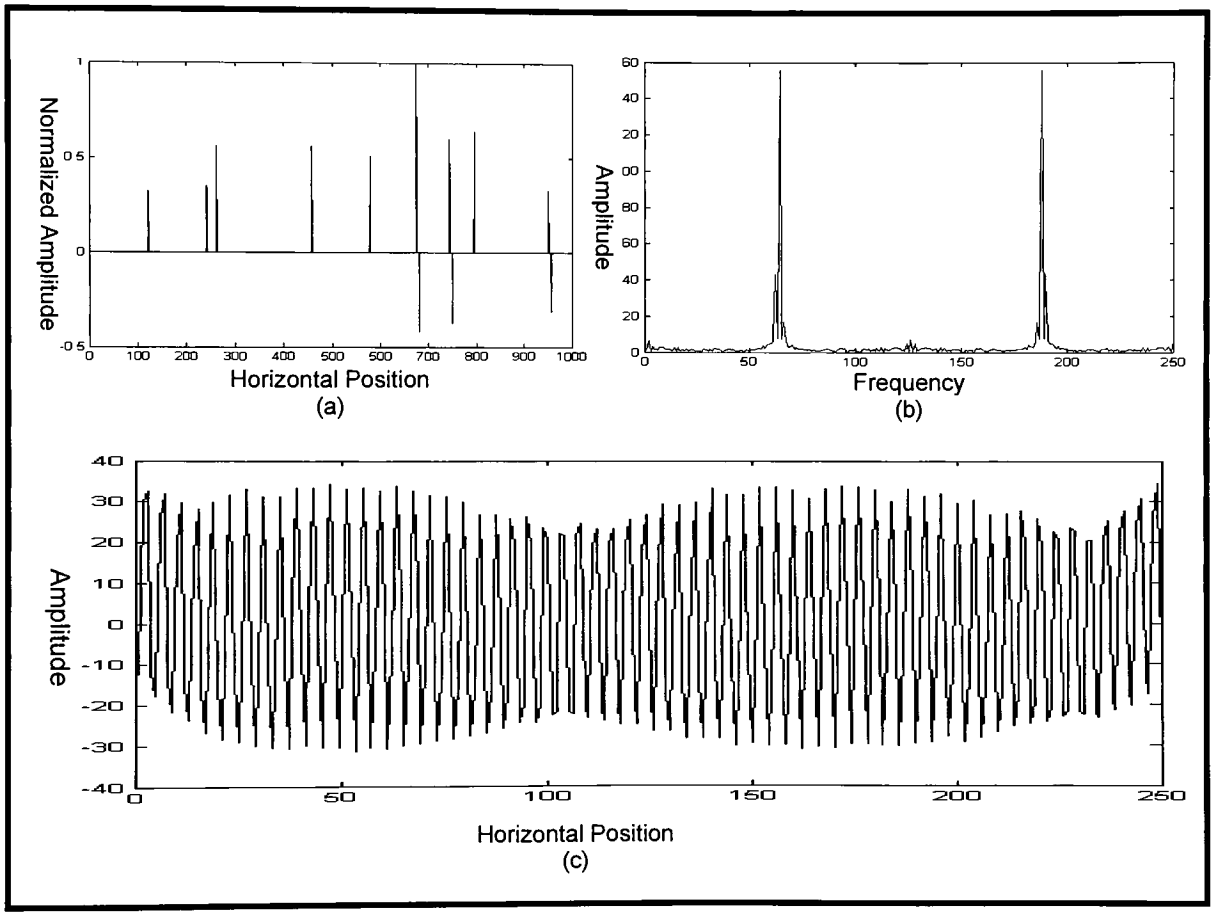


Figure 14: (a) Non Periodic Vertical Confidence Vector, (b) Periodic Vertical Confidence Vector Fourier transform, (c) Periodic Vertical Confidence Vector.

3.4 Computation of Length and Intensity of Streaks

To compute the length and intensity of a given vertical streak, this proposed algorithm starts by constructing a window that spans the streak horizontally and extends vertically

to the full image height (see Fig. 15a). The width of the window is defined by the width of the peak in the confidence vector. It then computes the vertical projection of the streak of interest for the selected window size. The height of the window is then reduced sequentially and the above process is repeated in a sliding scenario yielding a set of projections as a function of vertical window size (see Fig. 15b, 15c). The projection that yields the maximum or minimum value (dependent on whether the streak intensity is greater or smaller than the surrounding background) is utilized to determine the length and intensity thereby maximizing signal to noise ratio. To this effect, the height and average projection value of the window employed to compute the selected vertical projection is selected as the length and intensity of the streak, respectively.

The intensity and length of the streak are given by:

$$Intensity = \frac{1}{End - Start + 1} \sum_{j=Start}^{End} x(i, j) \tag{11}$$

$$k - \frac{\psi}{2} < i < k + \frac{\psi}{2}$$

Length of the streak=End-Start+1

where :

k : Horizontal position of the streak

ψ : Width of the streak

j =1,2,3…M (number of rows in the image), i =1,2,3,4….N (number of columns in the image).

End: Final position of the streak

Start: Initial position of the streak

$X(i, j)$: Pixel intensity value at position (i, j) .

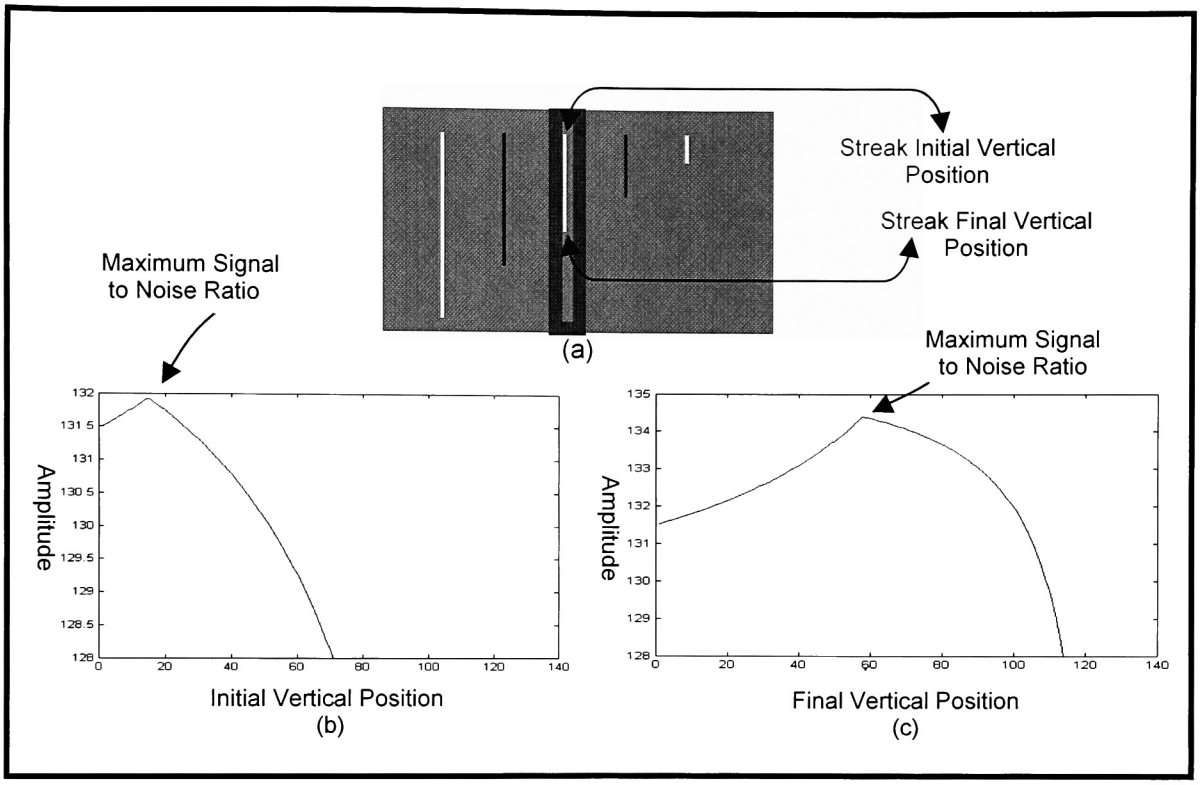


Figure 15: Determination of Length of the streak, (a) Spanning window, (b) Initial Vertical Position Profile, (c) Final Position Profile.

This focused approach is selected to minimize the number of computations that would be required for sliding window projections. The process is repeated in a similar fashion for horizontal type streaks.

Chapter 4

4.0 RESULTS

The performance of the proposed algorithm was tested on two sets of images. The first set consists of synthetic images that are constructed with various streak locations, lengths, width and intensities in order to test and quantify the robustness of the algorithm in the presence of noise. The second set encompasses several scanned images that contain varying degrees of streaks, noise, mottle, luminance gradient, and halftone periodic structure. These "real life" images were acquired from several electro-photographic based print engines. All hardcopies were scanned at 600 dpi yielding RGB color images. In the sections to follow, I will demonstrate the proposed approach for detecting vertical streaks

since the detection of horizontal streaks is identical assuming the image is pre-rotated by ninety degrees

4.1 Analysis on synthetic Images

Figure 16a shows a synthetic image created with a background gray value equal to 128 and different streaks lengths and intensity values as shown in Table 1. The streaks are numbered from left to right. Gaussian noise with different variances ranging from 1 to 90 (see Table 1) was added. The proposed algorithm was applied to the image shown in Fig. 16a and the resulting streaks and final confidence vector, for $\sigma^2 = 20$, are shown in Figure 13b and 13c respectively. The detected streak lengths are compared using absolute error to the streaks manually segmented by a human operator, and the outcome is displayed in Table 1. Note that the algorithm was able to detect the location of the streaks effectively with the exception of streak #6 which is 5 pixels in length. Hence, the ability to detect streaks is inversely proportional to the level of noise added as seen in Table 1 and depends heavily on the streak length. The longer the streak, the more likely it is to be detected in the presence of noise due to the increase in signal to noise ratio.

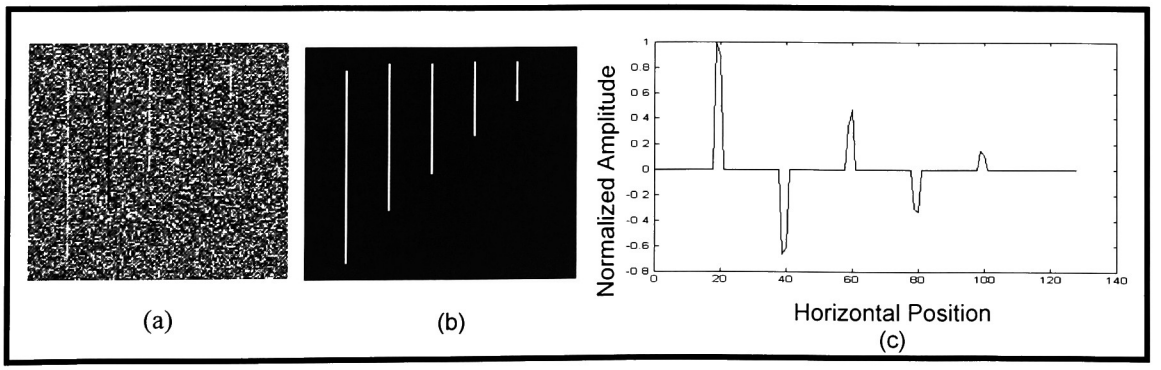


Figure 16: Synthetic image with streaks of different length and intensity, (b) Detected streaks for $\sigma^2 = 20$, (c) Confidence Vector.

Table 1: Length vs Noise Variance

-: Indicates that the streak was not detected

Streak Information				Absolute error as a function of Noise level (σ^2)				
No.	Length	Intensity	Width	1	20	40	60	90
1	110	136	1	0	1	1	2	2
2	80	120	1	0	2	2	2	2
3	60	136	1	0	2	2	2	-
4	40	120	1	0	3	3	10	-
5	20	136	1	0	0	-	-	-
6	5	120	1	-	-	-	-	-

Similarly, Figure 17 shows a synthetic image created with a background gray value equal to 128 and different streaks lengths, widths, and intensity values as shown in Table 2. Once again, the streaks are numbered from left to right and Gaussian noise with different variances ranging from 1 to 260 (see Table 2), instead of 1 to 90, was added. The proposed algorithm was applied to the image shown in Fig. 17a and the resulting streaks and final confidence vector, for $\sigma^2 = 20$, are shown in Figure 17b and 17c respectively. The detected streak lengths and widths are compared using absolute error to the streaks manually segmented by a human operator, and the outcome is shown in Table 2. Similar results were observed in this experiment with one exception. The widths of the streaks increased the likelihood of detection in the presence of noise as observed from Table 2.

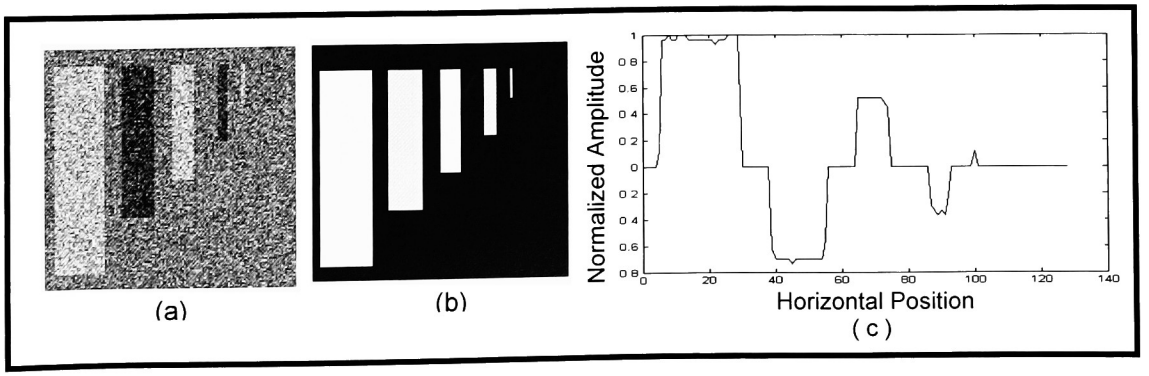


Figure 17: (a) Synthetic image with streaks of different width and intensity, (b) Detected streaks for $\sigma^2 = 20$, (c) Confidence Vector.

Table 2: Width vs. Noise Variance

- : Indicates that the streak was not detected

Streak Information				Absolute error as a function of Noise level (σ^2)				
No.	Length	Intensity	Width	1	40	120	200	260
1	110	136	25	0	0	1	1	-
2	80	120	15	0	0	0	1	-
3	60	136	10	0	1	1	2	-
4	40	120	5	0	1	1	2	-
5	20	136	2	0	-	-	-	-
6	5	120	1	-	-	-	-	-

Figure 18 shows a synthetic image created with a background gray value equal to 128 and several streaks of fixed length and width, and varying intensity values ranging from 134 to 150. The intensity values as shown in Table 3. Once again, the streaks are numbered from left to right and Gaussian noise with different variances ranging from 1 to 110 was added. Our proposed algorithm was applied to the image shown in Fig. 18a and the resulting streaks and final confidence vector, for $\sigma^2 = 20$, are shown in Figure 18b and 18c respectively. The detected streaks were compared using least mean square error to those manually segmented by a human operator. The results indicate that the probability of detection increased with the increase in intensity (see Table 3).

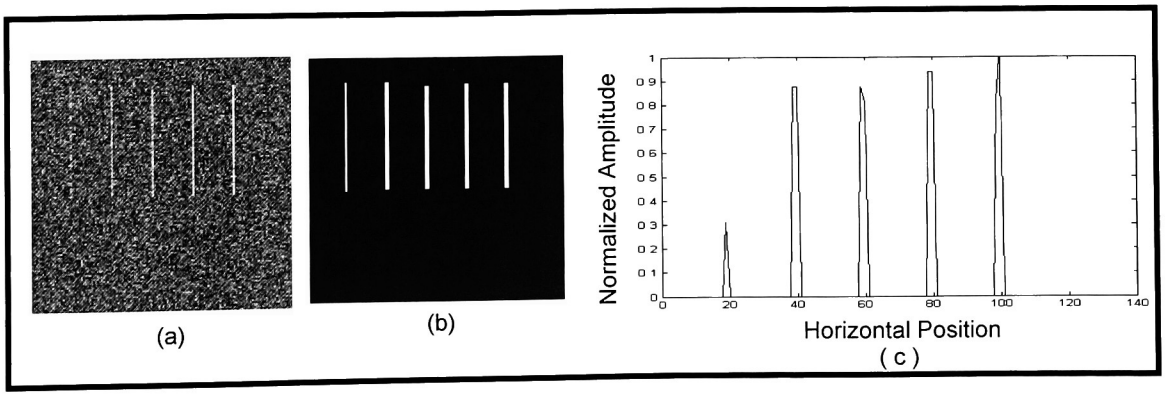


Figure 18: (a) Synthetic image with streaks of different intensity, (b) Detected streaks for $\sigma^2 = 20$, (c) Confidence Vector

Table 3: Intensity vs Noise Variance
-: Indicates that the streak was not detected

Streak Information				Absolute error as a function of Noise level (σ^2)				
No.	Length	Intensity	Width	1	20	40	60	110
1	55	134	1	2	2	-	-	-
2	55	138	1	2	1	2	6	-
3	55	142	1	1	1	2	2	1
4	55	146	1	1	1	1	1	1
5	50	150	1	1	1	1	1	1

4.2 Analysis on Real Images

In addition to the synthetic images discussed above, the algorithm was tested on several real life RGB scanned images that contain mottle, noise, halftone structures, luminance gradient and varying degrees of streaks length, width, and intensity. The threshold utilized to detect the location of the streaks was derived using a receiver operating characteristic (ROC) curve scenario as described in detail in [14, 15]. In summary, a set of 8 training images were selected and varied the threshold from 0.2 to 1.0. For each threshold value, the probability of detection and false alarm respectively was computed. The optimum threshold was chosen to maximize the probability of detection while

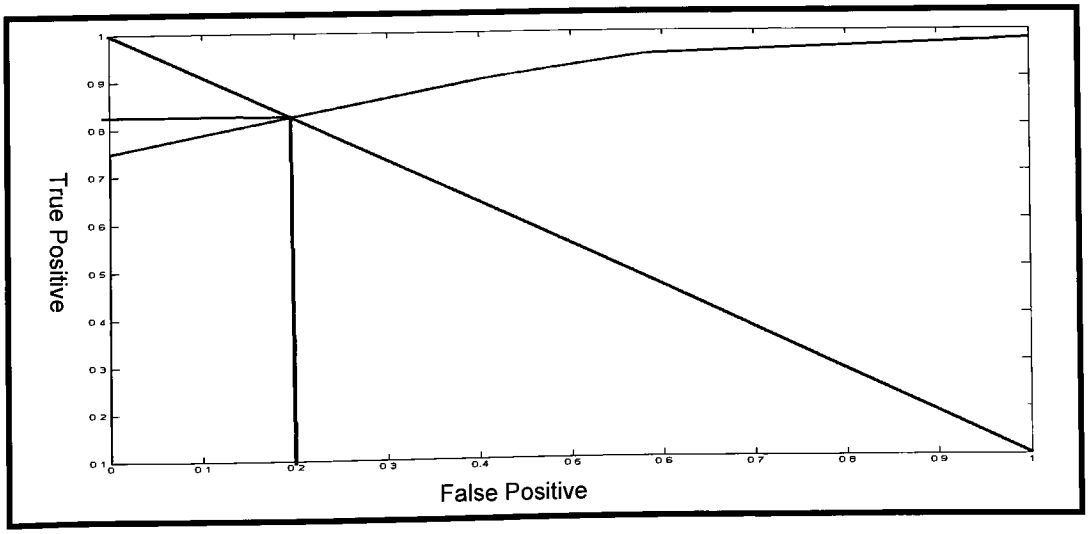


Figure 19: ROC analysis curve

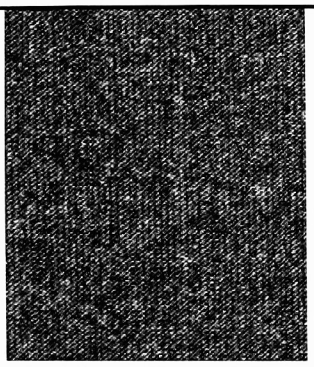
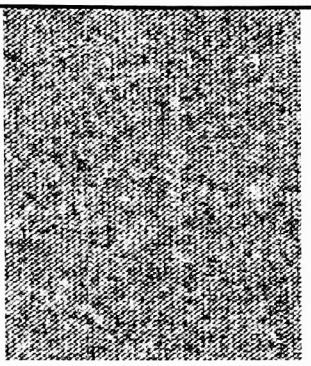
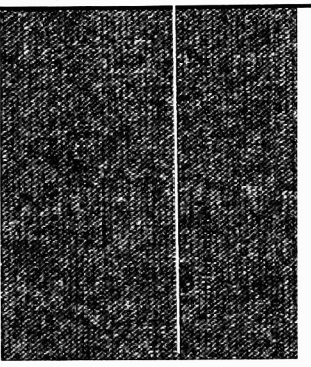
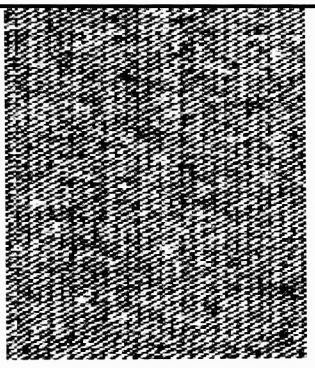
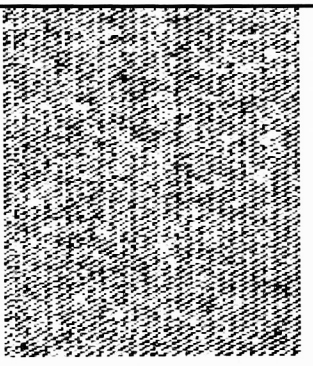
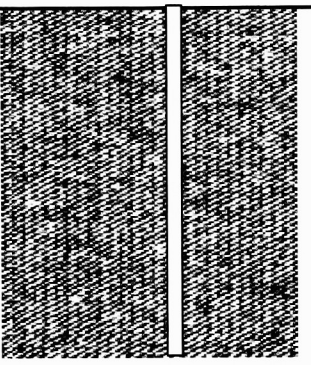
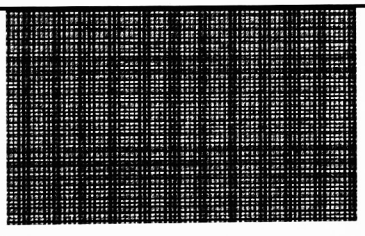
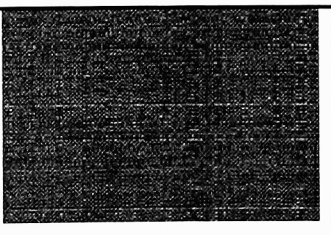
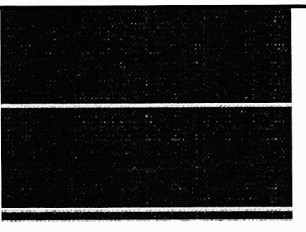
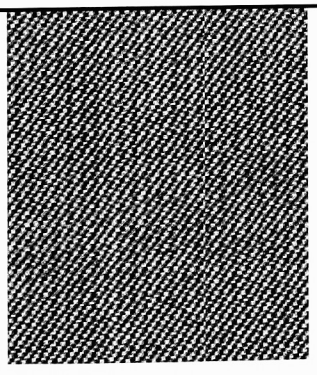
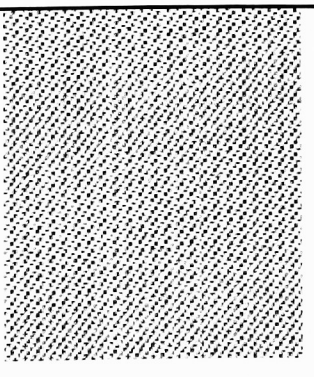
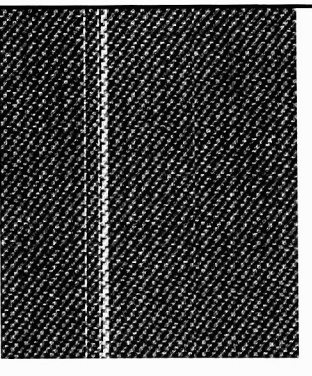
simultaneously minimize the probability of false alarm between the automatically detected streaks and the corresponding human segmented gold standard (see Fig 19).

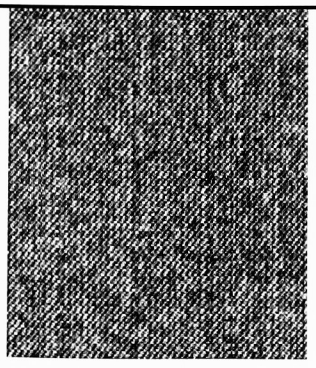
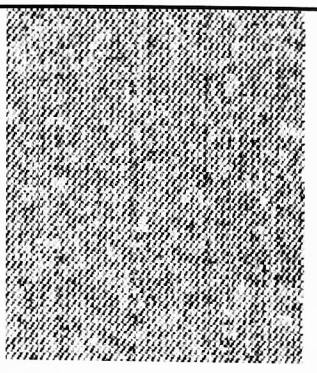
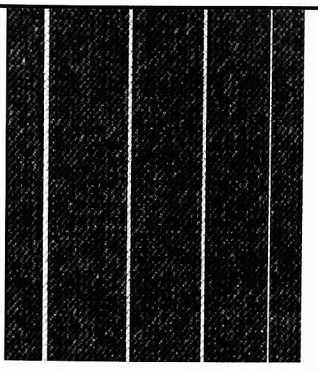
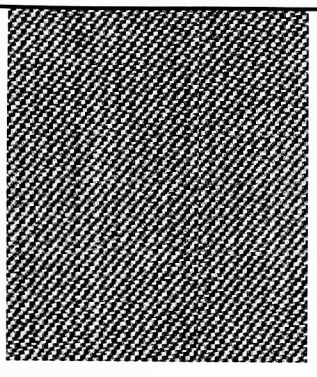
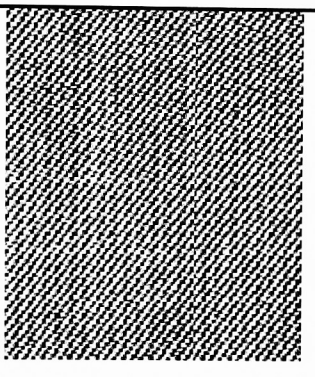
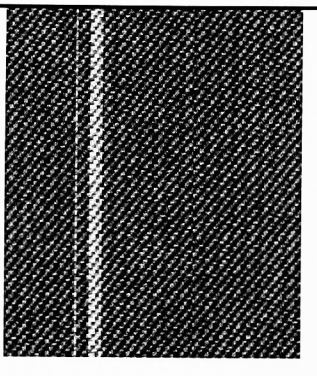
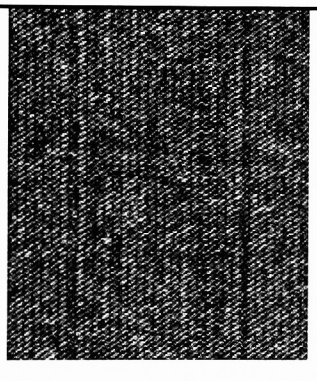
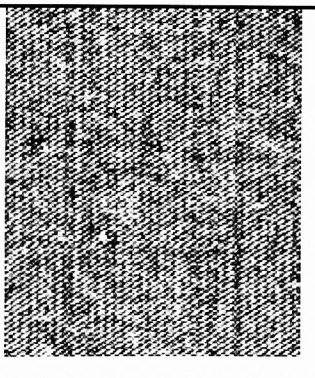
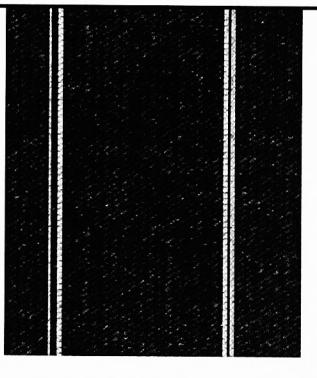
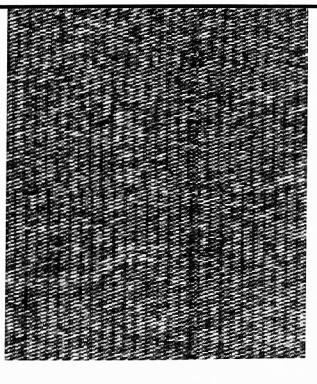
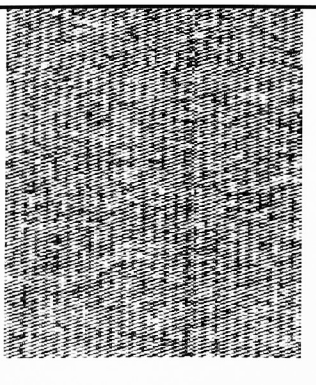
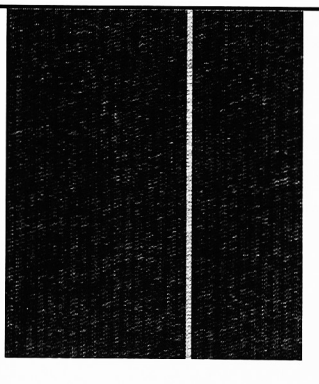
The average of optimum thresholds from the training set was chosen as the optimum threshold used to detect streaks on the real life images (see Table 4).

Table 4: Optimum Thresholds

Training images id	Optimum threshold
1	0.30
2	0.28
3	0.18
4	0.18
5	0.20
6	0.18
7	0.18
8	0.15
Average	0.21

Figure 20 shows the results obtained using different input real life images. In the figure, the highest contrast channel is shown (left), and the resulting streaks superimposed on the (right). An image enhancement method to the highest contrast channel images was applied for better visualization of the streaks in the input images. This is not a step of the proposed algorithm. The results or output images are "blow-up" of a region from different samples. The superimposed white lines indicate the location of the major streaks found in each of the images. The results display four different major types of streaks: I) a wide streak (Image id. 2 and 4) ii), single or several narrow streaks (Image id.1, 4, 5, 7, 8), iii) a set of streaks embedded in a periodic halftone structure (Image id 3), and 1v) periodic streaks (image id 9). The results show some of the biggest streaks for legibility. The length, width and intensity of the streaks found for the images 1 through 9 are tabulated in Tables 5, 6 and 7.

Image id	Highest Contrast Channel	Enhanced Image	Output Image
1			
2			
3			
4			

5			
6			
7			
8			

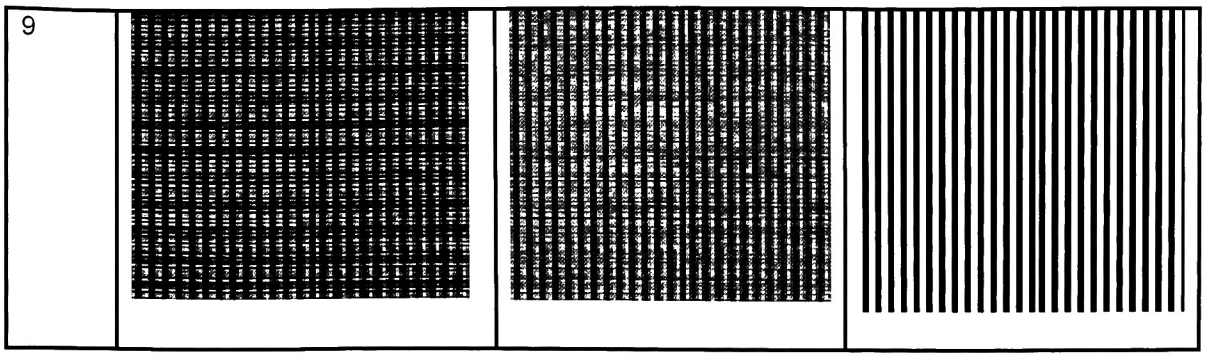


Figure 20: Results on Real Images

The results obtained show clearly that the algorithm was able to detect streaks on images that contain different degrees of mottle, noise, halftone structures, gradient in luminance and varying degrees of streaks length, width and intensity. The algorithm presented better results detecting larger streaks. This is due to the fact that these streaks present larger signal to noise ratio. For small streaks the algorithm presented some problems for their detection. Small streaks present a low signal to noise ratio. A low signal to noise ratio in presence of noise, mottle and gradient in luminance made difficult their detection.

Table 5: Length
-: No streak detected

Im. id	Streak id			
	1	2	3	4
1	222		-	
2	110	110		
3	114	114		
4	151	151		
5	139	139	139	139
6	143	143	-	
7	129	129	129	129
8	110			
9	100	100	(Periodic)	

Table 6: Width
-: No streak detected

Im. id	Streak id			
	1	2	3	4
1	2		-	-
2	3	1		
3	1	1		
4	1	3		
5	3	2	2	1
6	1	6		
7	1	2	2	2
8	2	-		
9	1	1		

Table 5: Intensity
-: No streak detected

Im id	Streak id			
	1	2	3	4
1	122			
2	118	113		
3	118	114		
4	75	74		
5	122	118	116	120
6	75	83		
7	110	116	117	110
8	110	-		
9	84	102		

Chapter 5

5. CONCLUSION

This thesis presented a new method for detecting streaks in mottled and noisy images by utilizing adaptive window-based image projections and maximization of mutual information. The traces collected from the projections are correlated using maximization of mutual information to pin-point horizontal position and width of the streak (s). For a given peak in the confidence vector (position and width) adaptive window sizes are used while maximizing the signal to noise ratio to find length and intensity. The proposed algorithm has been successfully demonstrated on a series of synthetic and real-life images that contain varying degrees of noise, mottle, and streaks. From the experimental results, I can conclude the following:

1) Streak detection is inversely proportional to the level of noise added and depends heavily on the streak length. As the noise increased smaller streaks tended not to be detected by the algorithm (see Table 1).

2) The widths of the streak increase the likelihood of detection in the presence of noise (see Table 2).

These conclusions are based on the fact that longer and wider streaks have larger signal to noise ratio which makes their detection easier for the algorithm. It can detect periodic and non periodic streak (s) due to the incorporation of Fourier transform on the confidence vector.

REFERENCES

[1] A. Kak and M. Slaney, "Principles of Computerized Tomographic Imaging", IEEE Press, New York, 1987.

[2] J. Hsieh, "Adaptive Streak Artifact Reduction in Computed Tomography Resulting From Excessive X-ray Photon Noise", *Med. Phys.*, Vol. 25, pp. 2139–2147, 1998.

[3] M. Kachelreiß, O. Watzke, and W. A. Kalender, "Generalized Multidimensional Adaptive Filtering For Conventional And Spiral Single-Slice, Multislice, And Cone Beam CT", *Med. Phys.*, Vol. 28, pp. 475–490, 2001.

[4] P. J. La Riviere and D. M. Billmire, "Reduced of Noise-Induced Streak Artifacts in X-Ray Computed Tomography Through Spline-Based Penalized-Likelihood Sinogram Smoothing", IEEE Trans. on Medical Imaging, Vol. 24 No. 1, January 2005.

[5] R. P. Loce, W. L. Lama and M. S. *Maltz,* "Modeling Vibration-Induced Halftone Banding in a Xerographic Laser Printer", Journal of Electronic Imaging, vol. 4, No. 1, pp. 48-61, January 1995.

[6] H. Kawamoto, K. Udagawa and M. Mori, "Vibration and Noise Induced by Electrostatic Force *on* a Contact Charger Roller of Electrophotography", Joamal of Image Science and Technology, Vol. 39, No. 6, pp. 477480, December 1995.

[7] H, Kawamoto, "Chatter Vibration of a Cleaner Blade in Electrophotography", Journal of Image Science and Technology, Vol. 40, m. 1, m. 8-13. January/February 1996.

[8] C. L. Chen and G.T. C. Chiu, "Banding Reduction in Electrophotographic Process", International Conference on Advanced Intellegent Mechatronics Proceedings, July 2001.

[9] D. R. Rasmussen, E. N. Dalal, and K. M. Hoffman, "Measurement of Macro-Uniformity: Streaks, Bands, Mottle and Chromatic Variations", PICS 2001: Image Processing, Image Quality, Image Capture Systems Conference, Montreal, Quebec, Canada; 2001.

[10] P Viola, "Alignment by Maximization of the Mutual Information", Technical Report 1548, June 1995.

[11] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Prentice Hall, 2002.

[12] G. Tourassi, E. Frederick, M. Markey, and C. Floyd, Jr., "Application of the Mutual Information for Feature Selection in Computer-Aided Diagnosis", Medical Physics, Vol. 28, Dec. 2001.

[13] C. O daub, R. Steuer, J. Selbig, and S. Kloska, "Estimating Mutual Information Using B-Spline Functions- an Improved Similarity Measure for Analyzing Gene Data", BMC Bioinformatics 2004, 5:118.

[14] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers", Kluwer Academic Publishers, Netherlands, March 2004.

[15] E. Saber, A. M. Tekalp, R. Eschbach, and K. Knox, "Automatic Image Annotation Using Adaptive Color Classification", Graphical Models and Image Processing, Vol. 58, No. 2, March, pp. 115-126, 1996.